Evolving Robot Morphology Facilitates the Evolution of Neural Modularity and Evolvability

Josh C. Bongard
Dept. of Computer Science
University of Vermont
josh.bongard@uvm.edu

Anton Bernatskiy
Dept. of Computer Science
University of Vermont
anton.bernatskiy@uvm.edu

Ken Livingston
Dept. of Cognitive Science
Vassar College
livingst@vassar.edu

Nicholas Livingston
Robotics Laboratory
Vassar College
nilivingston@vassar.edu

John Long
Dept. of Biology
Vassar College
jolong@vassar.edu

Marc Smith
Dept. of Computer Science
Vassar College
mlsmith@vassar.edu

ABSTRACT

Although recent work has demonstrated that modularity can increase evolvability in non-embodied systems, it remains to be seen how the morphologies of embodied agents influences the ability of an evolutionary algorithm to find useful and modular controllers for them. We hypothesize that a modular control system may enable different parts of a robot's body to sense and react to stimuli independently, enabling it to correctly recognize a seemingly novel environment as, in fact, a composition of familiar percepts and thus respond appropriately without need of further evolution. Here we provide evidence that supports this hypothesis: We found that such robots can indeed be evolved if (1) the robot's morphology is evolved along with its controller, (2) the fitness function selects for the desired behavior and (3) also selects for conservative and robust behavior. In addition, we show that if constraints (1) and (3) are relaxed, or structural modularity is selected for directly, the robots have too little or too much modularity and lower evolvability. Thus, we demonstrate a previously unknown relationship between modularity and embodied cognition: evolving morphology and control such that robots exhibit conservative behavior indirectly selects for appropriate modularity and, thus, increased evolvability.

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Artificial Intelligence-Robotics

Keywords

Artificial Intelligence, Robotics, Evolutionary Computation, Modularity, Evolvability, Embodied cognition, Artificial neural networks

GECCO '15, July 11–15, 2015, Madrid, Spain

DOI: http://dx.doi.org/10.1145/2739480.2754750

1. INTRODUCTION

In the field of evolutionary robotics [14, 10, 2], an evolutionary algorithm is typically employed to optimize the wiring diagram of the robot's neural controller and/or aspects of its morphology such that the robot exhibits some desired behavior. However, for complex robots with large numbers of artificial neurons, the number of possible connections between those neurons grows quadratically, requiring search over a very large space of candidate controllers. Another barrier to continued evolutionary improvement is that large, densely-connected neural controllers are notoriously difficult to evolve: any mutation to one part of the network tends to affect behavior at many other parts of the network.

Both of these reasons suggest that it would be desirable to evolve robots controlled by modular neural networks. However, it is unclear how to architect an evolutionary algorithm to evolve controllers with modularity suitable to the task the robot must perform.

Previous work has investigated evolving modular control if the task calls for it. In [13] however, the number and type of modules, as well as the connectivity pattern, were predetermined. Gruau [8] employed an indirect genotype to phenotype mapping that allowed for the construction of neural modules. However, this method requires an evolutionary operator for explicitly creating modules. In other work [3] an evolutionary operator could duplicate existing modules. The work presented here differs from all of these approaches in that it does not require such high-level operators: modules emerge (or dissipate) as local connectivity density gradually increases (or decreases) over evolutionary time.

The challenge of evolving modularity without pre-supposing the cognitive architecture of the controller, or explicitly having to formulate module-creating operators, is demonstrated in [4]. There, it was shown that HyperNEAT [17], a popular indirect encoding method, failed to evolve neural modularity on all but the simplest of problems.

Outside of evolutionary robotics, many recent projects have investigated the conditions under which structural modularity will evolve[1]. However, these approaches have investi-

[1]Structural modularity—in which dense connectivity exists within modules and few connections exist between modules—is distinguished from functionally modularity[20],

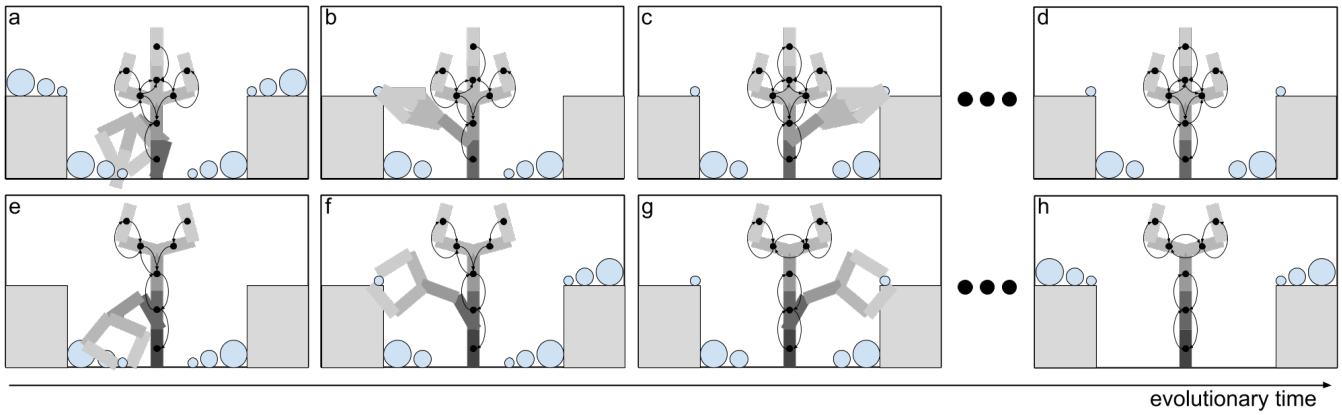

Figure 1: The hypothesized relationship between morphology, neural modularity, and evolvability. Two robot arms with different morphologies are evolved to move objects of different size onto two ledges: one arm supports a three-fingered gripper (a-d) while the other supports a two-fingered gripper (e-h). Both evolve the ability to grasp one object (a,e) and place it somewhere else (b,f). Later in evolutionary time, both evolve the ability to place two of the objects (c,g). During this second evolutionary phase, the second robot evolves a modular neural controller: synapses connect gripper motor neurons and other synapses connect the arm motor neurons, but there are no synapses connecting gripper and arm neurons (g). The first robot does not evolve neural modularity: its gripper and arm motor neurons do not become dissociated (c). This modularity renders the second arm more evolvable than the first such that it eventually evolves the ability to place all six objects (h).

gated modularity in a non-embodied context: in [9] and [5], a disembodied visual system must recognize combinations of familiar and unfamiliar patterns; in [6], gene regulatory networks were modeled without taking into account the effect that the cell's environment has on gene regulation.

In contrast, here we investigate the evolution of modular controllers in embodied agents and demonstrate a hitherto unknown relationship between modularity and adaptive behavior: evolving robot morphology along with control can lead to modular controllers which, in turn, increase the overall evolvability[2] of the population; however, if the robots' body plans are fixed, this evolvability cannot be achieved.

The robots evolve modular controllers because they experience a combination of directional selection and stabilizing selection. This combination of evolutionary pressures has been implicated as the driving force behind the evolution of modularity in biological organisms [19]. More recently, this theory was supported experimentally by observing the evolution of models of gene regulatory networks under these conditions [6].

In the same work [6], the link between modularity and evolvability was made clear: modular networks can rapidly produce novel combinations of previously- evolved gene activity patterns. Here, we demonstrate that robots with modular controllers are evolvable for similar reasons: such robots can recognize seemingly novel situations as novel combinations of familiar percepts. They are thus able to respond appropriately without having to evolve a novel response to this seeming novel situation.

in which a network exhibits several distinct functional pattens.

[2]Here we define the evolvability of a system as the average increase in fitness that can be achieved, given a fixed computational budget.

2. HYPOTHESIS

Our hypothesis is that robots with certain morphologies will experience directional selection acting on one part of their bodies and controllers, and stabilizing selection acting on another part of their bodies and controllers, as they evolve to perform some desired task. This combination of selection pressures will favor neural modularity that will in turn enable them to rapidly adapt when presented with novel combinations of familiar percepts: only slight evolutionary modification will be required in one or a few neural modules.

Fig. 1 provides a visual outline of the hypothesis we wish to test. Two robots with different morphologies evolve to perform an object pick-and-place task. The first robot's morphology does not experience the correct combination of selection pressures (Fig. 1a-b), does not evolve neural modularity (Fig. 1c), exhibits low evolvability, and fails to become increasingly adept at the task (Fig. 1d). The second robot happens to have the right morphology (Fig. 1e-f) that eventually leads to neural modularity (Fig. 1g) and increased evolvability (Fig. 1h).

Below, we demonstrate that this relationship between morphology, neural modularity and evolvability does indeed exist, at least for a simplified robot and task environment.

3. METHODS

This section outlines the robots, their task environments, their controllers, and the evolutionary algorithm we used to evolve them. All of the material for replicating the work described here is available at `http://git.io/vvH7r`.

3.1 The Robots

The robots used in this work are two dimensional, are composed of a three-segment arm (black lines in Fig. 2) and a hand composed of two, two-segment fingers (red and

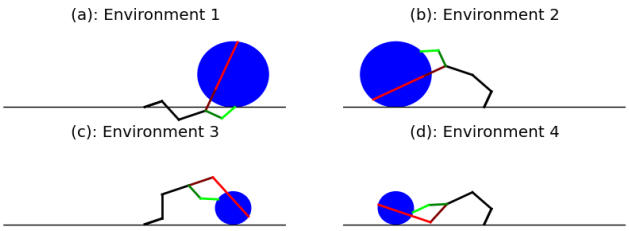

Figure 2: A typical evolved non-conservative pose. The robot's arm (black segments) and hand (red and green segments) reach a different pose in each environment.

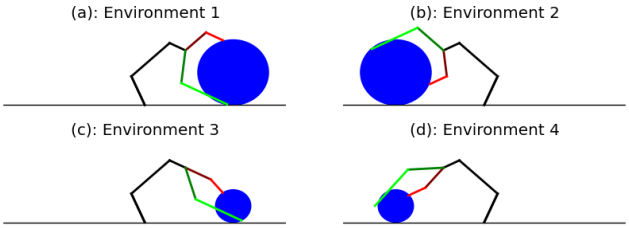

Figure 3: A typical evolved conservative pose. The hand reaches the same pose in environments 1 and 2, and environments 3 and 4. The arm reaches the same pose in environments 1 and 3, and environments 2 and 4.

green lines in Fig. 2), and are based on the robots investigated in [1]. Each of the seven segments can vary in length between $0.5u$ and $3.0u$[3]. Segments are attached by rotational hinge joints, each of which can take on one of four possible angles. The three arm joints can take any angle in $\{-70^o, -23.33^o, +23.33^o, +70^o\}$: negative and positive angles rotate the arm to the left and right, respectively. The left proximate phalange and right distal phalange can take any value in $\{-70^o, -46.67^o, -23.33^o, 0^o\}$; The right proximate phalange and left distal phalange can take any value in $\{0^o, 23.33^o, 46.67^o, 70^o\}$; These angle constraints for the hand ensure that it maintains claw-like poses and that the phalanges do not cross.

In the experiments in which the robot's morphology is evolved, a seven-element, floating-point valued vector $\mathbf{m}$ was evolved: element m_i in this vector encodes the length of the ith segment.

3.2 The Task Environments

The robots were evolved to grasp objects in four different environments. The first environment contained an object with radius $1u$ placed $2.5u$ to the right of the robot's base. The second environment contained an object with radius $1u$ placed $2.5u$ to the left of the robot's base. The third environment contained an object with radius $0.5u$ placed $2.5u$ to the right of the robot's base. The fourth environment contained an object with radius $0.5u$ placed $2.5u$ to the left of the robot's base. No collision detection and resolution was employed: the robot's segments could pass through the object. The robot and object were simulated in a kinematic simulation.

[3] u is a unitless measure of distance.

3.3 The Controllers

The robots were evolved with either a simple or random boolean network (RBN) controller.

Simple Control. The simple controller was encoded as a 4×7 matrix $\mathbf{C}$, in which the ith row encoded joint angles for the ith environment and the j column encoded angles for the jth joint. When the robot was 'controlled' by this simple controller, its joints were simply set to these angles and forward kinematics were used to compute the resulting pose of the robot. Fig. 2 displays a typical pose when the robot is evolved to grasp the four objects using this controller.

RBN Control. The RBN controller is composed of 14 binary neurons that can take on values in $\{-1, +1\}$. Each of the seven joints is assigned two of these neurons. The first six neurons are assigned to the three arm joints; the last eight neurons are assigned to the four hand joints. The four possible neuron value pairs $[-1, -1]$, $[-1, +1]$, $[+1, -1]$, and $[+1, +1]$ for each joint dictate which of its four possible angles it rotates to. Thus, each of the $2^{14} = 16,384$ possible states of the controller corresponds to a unique robot pose. Each controller is encoded as a 14×14 adjacency matrix $\mathbf{A}$. Elements in this matrix are trinary: $a_{ij} = +1$, $a_{ij} = -1$, and $a_{ij} = 0$ indicate that neuron i is connected to node j with an excitatory synapse, inhibitory synapse, or no synapse, respectively.

When a robot is placed into one of the four environments, the initial values of the neurons are set to indicate the size and position of the object in that environment. Thus, the robot can 'sense' its surroundings based on the neurons' initial values. Even-numbered neurons indicate object size: $n_0, n_2, \ldots, n_{12}$ are all set to $+1$ if the object is large and to -1 if the object is small. Odd-numbered neurons indicate object position: $n_1, n_3, \ldots, n_{13}$ are all set to $+1$ if the object is on the robot's right and to to -1 if the object is on the robot's left.

After setting the initial conditions, the RBN is updated three times using synchronous updating. The value of the ith neuron at time step t is set to

$$n_i^{(t)} = \sigma\left(\sum_{j=1}^{14} a_{ji} n_j^{(t-1)}\right) \quad (1)$$

where

$$\sigma(x) = \begin{cases} -1, & \text{if } x <= 0 \\ +1, & \text{otherwise} \end{cases} \quad (2)$$

The final values of the neurons are then used to set the robot's pose as described above. Fig. 3 displays a pose produced by an evolved RBN controller.

3.4 The Evolutionary Algorithm

We employed the Age-Fitness Pareto Optimization (AFPO) evolutionary algorithm [16] for all of the experiments reported here. AFPO is a multiobjective optimization method that periodically injects new genomes into the evolving population, and enables these younger genomes time to improve before being outcompeted in the population by older genomes with higher fitness. This enables older genomes trapped on local optima to periodically be displaced by younger genomes that, with time, evolve to higher fitness values. AFPO achieves this dynamic using the genome's age as one fitness objective.

Aside from age, genomes were also evolved to produce robots that could grasp objects, had modular controllers, and/or exhibited behavioral conservatism.

Grasping ability. Evolving robots could be exposed to one or more of the four environments shown in Fig. 2. In each environment, grasping ability was determined by how close the robot could get the tips of its fingers to the circumference of the object, and ensure that the fingertips touched on opposite sides of the objects.

In addition, the robot may have to exhibit grasping in the face of a perturbation: in some evaluations, after the initial values of the neurons have been set, one of the binary neuron values is flipped from $+1$ to -1 (or -1 to $+1$). The robot is expected to still exhibit successful grasping in the face of this perturbations.

A robot's grasping ability g was computed using

$$g = \sum_{r=0}^{14} \sum_{s=1}^{2} \sum_{p=1}^{2} g_{rsp}, \quad (3)$$

$$g_{rsp} = g_{rsp}^{(L)} \; g_{rsp}^{(R)} \; g_{rsp}^{(S)}, \quad (4)$$

$$g_{rsp}^{(L)} = \left(\frac{1}{1 + d_{rsp}^{(L)}} \right), \quad (5)$$

$$g_{rsp}^{(R)} = \left(\frac{1}{1 + d_{rsp}^{(R)}} \right), \quad (6)$$

$$g_{rsp}^{(S)} = \left(\frac{1}{1 + d_{rsp}^{(S)}} \right), \quad (7)$$

$$d_{rsp}^{(S)} = |s_{rsp} - o_s|, \quad (8)$$

$$s_{rsp} = \sqrt{(x_{rsp}^{(L)} - x_{rsp}^{(R)})^2 + (y_{rsp}^{(L)} - y_{rsp}^{(R)})^2} \quad (9)$$

where

- $g = 0$ indicates that the robot failed completely in every environment, and $g = 15 \times 2 \times 2 = 60$ indicates that the robot grasped every object perfectly in every environment.
- g_{rsp} indicates the robot's ability to grasp during the rth perturbation, the sth object size, and the pth object position.
- For perturbation $r = 0$, no perturbation of the neurons' initial values in the robot's controller occurs. For $r = 1$ through $r = 14$, the initial value of the rth neuron is flipped.
- $s = 1$ indicates that the robot is in an environment in which the object is large; $s = 2$ indicates the object is small.
- $p = 1$ indicates that the robot is in an environment in which the object is on its right; $p = 2$ indicates the object is on its left.
- $g_{rsp} = 0$ indicates that the robot grasped the object in that environment poorly; $g_{rsp} = 1$ indicates it grasped it perfectly.
- $g_{rsp}^{(L)}$ (or $g_{rsp}^{(R)}$) indicate how close the robot's left fingertip (or right fingertip) got to the object's circumference, respectively.
- $d_{rsp}^{(L)}$ (or $d_{rsp}^{(R)}$) indicate the shortest distance between the robot's left fingertip (or right fingertip) and the object's circumference, respectively.
- $g_{rsp}^{(S)}$ denotes the spread of the robot's grasp: $g_{rsp}^{(S)} = 0$ indicates the robot's fingertips are touching the object's circumference at the same point (a poor grasp); $g_{rsp}^{(S)} = 1$ indicates the robot's fingertips are touching the object on opposite sides of its circumference (a good grasp).
- $d_{rsp}^{(S)}$ is the difference between the distance of the robot's fingertips—their spread (s_{rsp})—and the circumference of the sth object (o_s).
- Finally, $x_{rsp}^{(L)}$ and $y_{rsp}^{(L)}$ (and $x_{rsp}^{(R)}$ and $y_{rsp}^{(R)}$) denote the horizontal and vertical position of the robot's left (and right) fingertips.

During evolution, each robot was only exposed to the next environment if it achieved sufficiently good grasping in the current environment. The robot was allowed to behave in the next environment—and, thus, obtain higher fitness—if $g_{rsp} > 0.9$ and the robot's controller had settled into a fixed-point attractor. This latter stipulation ensures that the robot has not simply managed to bring its fingertips near the object during the third and final update of its controller but may move away from the object on subsequent updates.

Thus, more fit robots had to exhibit competency in a growing number of environments. Furthermore, as shown by the summation sequence in Eqn. (3), robots experienced changes in their environment in a particular order: they experienced a novel object position in environment 2 (Fig. 2b) and a novel object size in environment 3 (Fig. 2c).

Once a robot evolves to succeed in all four environments, it is evaluated again in the first environment, but now suffers a perturbation: the initial value of its first neuron is flipped. If successful there, the robot is evaluated a sixth time in environment 2 with a flipped first neuron value, and so on. Robots can thus be exposed up to a maximum of $15 \times 2 \times 2 = 60$ environments.

Controller modularity. The modularity of an evolved robot's controller was computed using

$$m = \frac{1 + d_{aa} + d_{hh}}{1 + d_{ah} + d_{ha}}, \quad (10)$$

$$d_{aa} = \frac{\sum_{a1=1}^{3} \sum_{a2=1}^{3} |a_{a1a2}|}{9}, \quad (11)$$

$$d_{hh} = \frac{\sum_{h1=4}^{7} \sum_{h2=4}^{7} |a_{h1h2}|}{16}, \quad (12)$$

$$d_{ah} = \frac{\sum_{a1=1}^{3} \sum_{h1=4}^{7} |a_{a1h1}|}{12}, \quad (13)$$

$$d_{ha} = \frac{\sum_{h1=4}^{7} \sum_{a1=1}^{3} |a_{h1a1}|}{12} \quad (14)$$

where d_{aa}, d_{hh}, d_{ah}, and d_{ha} denote the density of connectivity between the arm neurons, between the hand neurons, between the arm and hand neurons, and between the hand and arm neurons, respectively.

Thus m, like the more general modularity metric Q [12], measures the ratio of connection density within candidate modules, compared to connection density across modules[4].

[4]The Q modularity score was not employed here because of its computational overhead.

In this work we seek to measure specific modularity in the robot's controller: evolutionary dissociation between the arm neurons and the hand neurons.

Behavioral conservatism. It has been hypothesized [19] and shown experimentally in non-embodied systems [6] that structural modularity evolves when a system experiences stabilizing selection on one part of its phenotype and directional selection on another part. Here, we wished to select for robots that experience this kind of evolutionary pressure. We did so by formulating a measure c that gauges the amount of conservatism in a robot's evolved behavior:

$$c = \frac{c_a + c_h + c_r}{3}, \tag{15}$$

$$c_a = \left(\sum_{r=0}^{14} \sum_{s=1}^{2} \sum_{p_1=1}^{2} \sum_{p_2=1}^{2} \sum_{a=1}^{6} e_{rsp_1p_2} \left(n_a^{(rsp_1)} = n_a^{(rsp_2)} \right) \right) / e_p, \tag{16}$$

$$e_p = \sum_{r=0}^{14} \sum_{s=1}^{2} \sum_{p_1=1}^{2} \sum_{p_2=1}^{2} \sum_{a=1}^{6} e_{rsp_1p_2}, \tag{17}$$

$$c_h = \sum_{r=0}^{14} \sum_{s_1=1}^{2} \sum_{s_2=1}^{2} \sum_{p=1}^{2} \sum_{a=7}^{14} e_{rs_1s_2p} \left(n_h^{(rs_1p)} = n_h^{(rs_2p)} \right) / e_s, \tag{18}$$

$$e_s = \sum_{r=0}^{14} \sum_{s_1=1}^{2} \sum_{s_2=1}^{2} \sum_{p=1}^{2} \sum_{a=7}^{14} e_{rs_1s_2p}, \tag{19}$$

$$c_r = \sum_{r=1}^{14} \sum_{s=1}^{2} \sum_{p=1}^{2} \sum_{i=1}^{14} e_{rsp} \left(n_i^{(0sp)} = n_i^{(rsp)} \right) / e_r, \tag{20}$$

$$e_r = \sum_{r=1}^{14} \sum_{s=1}^{2} \sum_{p=1}^{2} \sum_{i=1}^{14} e_{rsp} \tag{21}$$

where

- $c = 0$ when the robot exhibits a different pose in every environment and under every perturbation. (Fig 2 shows a robot with this minimal conservatism.)
- $c = 1$ when the arm does the same thing in all environments in which the object is at the same position, the hand always does the same thing in all environments in which the object is the same size, and the overall robot does the same thing in the same environment, regardless of whether its initial state was perturbed or not. (Fig. 3 shows an evolved robot that achieves this: the hand reaches the same pose in environments 1 and 2 and environments 3 and 4, and the arm reaches the same pose in environments 1 and 3 and environments 2 and 4.)
- c_a measures the amount of similarity across arm poses for environments in which the object is on the same side,
- c_h measures the amount of similarity across hand poses for environments in which the object is the same size, and
- c_r measures the robot's robustness: how similar is the robot's overall pose when it experiences a perturbation compared to the same environment in which it does not experience a perturbation.
- $e_{rsp_1p_2}$ is equal to 1 if the robot was evaluated in the two environments rsp_1 and rsp_2, and the object had the same position in those environments ($p_1 = p_2$), and is equal to zero otherwise.
- $n_a^{(rsp_1)} = n_a^{(rsp_2)}$ is equal to 1 if arm neuron n_a holds the same value in a pair of environments with equal object positions, and zero otherwise.
- e_p denotes the number of environments with equally-positioned objects that the robot was evaluated in.
- $e_{rs_1s_2p}$ is equal to 1 if the robot was evaluated in the two environments rs_1p and rs_2p, and the object had the same size in those environments ($s_1 = s_2$), and is equal to zero otherwise.
- $n_h^{(rs_1p)} = n_h^{(rs_2p)}$ is equal to 1 if hand neuron n_h holds the same value in a pair of environments with equal object sizes, and zero otherwise. Finally,
- e_s denotes the number of environments with equally-sized objects that the robot was evaluated in.
- e_{rsp} is equal to 1 if the robot was evaluated in environment rsp and is equal to zero otherwise.
- e_r denotes the number of environments the robot was evaluated in.
- Finally, $n_i^{(0sp)} = n_i^{(rsp)}$ is equal to 1 if the ith neuron obtains the same final value in an environment in which the initial value of neuron r experienced no perturbation, and the same environment in which the initial value of neuron r was flipped, and zero otherwise.

Population initialization. Each population was seeded with 100 randomly-generated genomes. For robots evolved with simple controllers (experiment set 1 in Table 1), each genome was made up of one set of morphological parameters $\mathbf{m}$ and control parameters $\mathbf{C}$. For robots with fixed morphologies and RBN controllers (experiment sets 2, 4, and 6 in Table 1), each genome was comprised of an RBN adjacency matrix $\mathbf{A}$. For robots with evolved morphologies and evolved RBN controllers (experiment sets 3, 5, and 7 in Table 1), each genome comprised a set of morphological parameters $\mathbf{m}$ and an RBN adjacency matrix $\mathbf{A}$.

Randomly-generated $\mathbf{m}$ vectors were filled with random values drawn from $(0.5, 3.0)$ with a uniform distribution. Randomly-generated $\mathbf{C}$ matrices were filled with random angles drawn from the four possible angles for each joint. Randomly-generated $\mathbf{A}$ matrices were filled with $(2 \times 14 = 28)$ positive or negative edges, following the convention established in [6].

Selection. In each generation, each new genome was evaluated on the robot in as many environments as it was able to succeed in. Then, the Pareto front was calculated using the genome's age, grasping ability, controller modularity and/or behavioral conservatism, as shown in Table 1. All dominated genomes were then discarded. The population was refilled by selecting a non-dominated genome at random, copying it, mutating it, and placing it in an empty slot until the population size reached 99. The final empty slot was filled with a new randomly-generated genome with an age of zero.

Mutation. When $\mathbf{m}$ was mutated, zero (0.33 probability), one (0.33 probability) or two (0.34 probability) segment

Table 1: Summary of experimental settings.

Experiment set	Controller type	Morphology evolved?	Popn. size	Num of generations	Num of replicates	First objective	Second objective	Third objective	Environments
1	Simple	yes	100	100,000	100	age	grasping		4
2	RBN	no	100	1,000,000	100	age	grasping		60
3	RBN	yes	100	1,000,000	100	age	grasping		60
4	RBN	no	100	1,000,000	100	age	grasping	modularity	60
5	RBN	yes	100	1,000,000	100	age	grasping	modularity	60
6	RBN	no	100	1,000,000	100	age	grasping	conservatism	60
7	RBN	yes	100	1,000,000	100	age	grasping	conservatism	60

lengths in **m** were mutated. The current value was replaced with a new random value drawn from $(0.5, 3.0)$ with a uniform probability.

When **C** was mutated, zero (0.33 probability), one (0.33 probability) or two (0.34 probability) angles in **C** were mutated. The current value was replaced with a new angle drawn from the four possible angles for that joint.

A was mutated with a bias toward two or three incoming edges, following [6]. Each neuron in a newly-copied controller was targetted for mutation with probability 0.05. If selected for mutation, the probability that an incoming synapse to n would be deleted is defined as

$$p(n) = \frac{4r_n}{4r_n + (14 - r_n)} \tag{22}$$

and the probability that a randomly-selected incoming edge will be added to n is defined as $q(n) = 1 - p(n)$. Here r_n denotes the in-degree of neuron n.

4. RESULTS

The settings for all of the experiments described in this section are summarized in Table 1.

In the first experiment, 100 trials were conducted in which the morphologies of robots were evolved along with their simple controllers. These robots were evolved only to grasp, did not experience perturbations, and each robot was exposed to all four environments. One hundred robots were collected by extracting the best robot from each trial. The resulting behavior for one of these robots is shown in Fig. 2. All 100 of these robots succeeded in all four environments: each achieved $g > 0.96$ (data not shown).

In the 100 trials comprising the second experiment, each trial was seeded with one of the 100 robots from the previous experiment. Then, RBN controllers were evolved in each trial to enable the fixed-morphology robot assigned to that trial to succeed at grasping in (up to) the 60 possible environments. The third experiment was conducted in the same manner, except that the trials began with randomly-generated morphologies, and those morphologies then evolved along with the controllers. The 100 best robots were collected from the second experiment and the third experiment; differences between their mean grasping ability (Fig. 4a), controller modularity (Fig. 4b), behavioral conservatism (Fig. 4c) and pose differences (Fig. 4d) are reported. (Pose difference is defined as the converse of behavioral conservatism: how different was the robot's arm pose in environments with objects in the same place; how different was the robot's hand pose in environments with objects of the same size; and how different was the robot's overall pose with or without initial perturbation.)

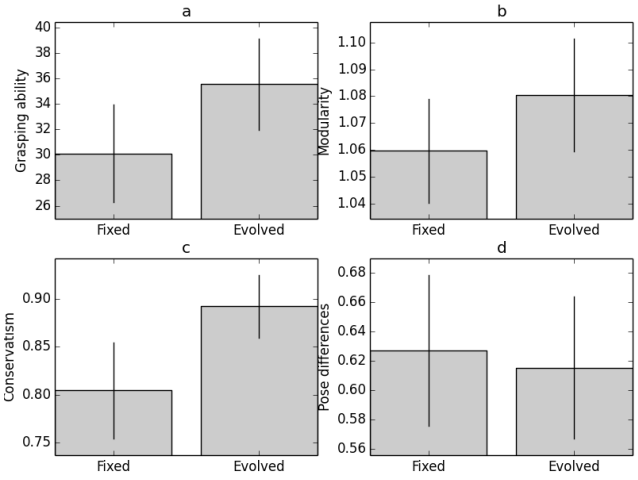

Figure 4: Relative performance of robots with fixed and evolved morphologies, when evolved to grasp the objects. Robots with evolved morphology exhibited (a) significantly better grasping ability (Mann-Whitney U-test, $p < 0.001$), (b) no significant difference in modularity, (c) significantly more behavioral conservatism (Mann-Whitney U-test, $p < 0.001$), and (d) no significant difference in the amount of pose differences. Error bars denote the 95% confidence intervals.

Two more experiments were conducted in which the robot's morphology was fixed or was allowed to evolve (experiment sets 4 and 5 in Table 1). In the 100 treatments involving a fixed-morphology robot (experiment set 4), the 100 morphologies evolved in experiment 1 were re-used. In these two experiments, the robots were evolved against the additional objective of controller modularity, in addition to grasping ability and genome age. The differences in mean grasping ability, controller modularity, behavioral conservatism and pose differences for the 100 best robots from each experiment are reported in Fig. 5.

A final pair of experiments were conducted (experiment sets 6 and 7 in Table 1) where, again, the robots' morphologies were held constant in the first set of 100 trials, and were allowed to evolve in the second set of 100 trials. The

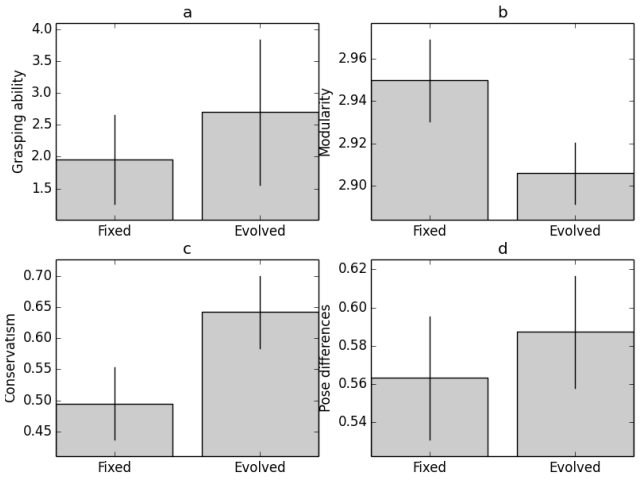

Figure 5: Relative performance of robots with fixed and evolved morphologies, when evolved to grasp the objects with highly modular controllers. Robots with evolved morphology exhibited (a) significantly better grasping ability (Mann-Whitney U-test, $p < 0.001$), (b) significantly lower modularity (Mann-Whitney U-test, $p < 0.001$), (c) significantly more behavioral conservatism (Mann-Whitney U-test, $p < 0.001$), and (d) no significant difference in the amount of pose differences.

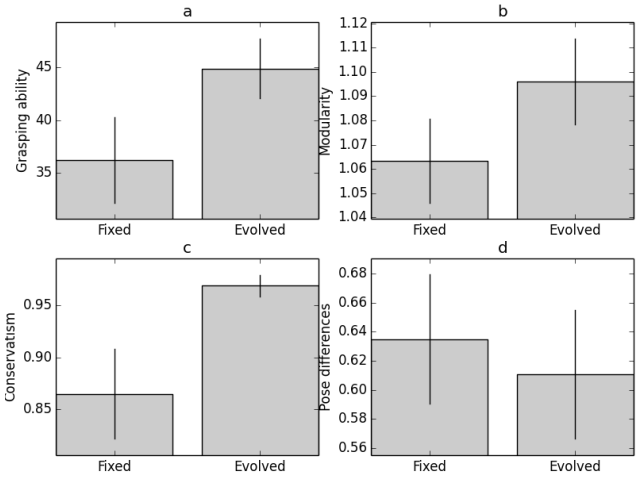

Figure 6: Relative performance of robots with fixed and evolved morphologies, when evolved to grasp the objects in a conservative manner. Robots with evolved morphology exhibited (a) significantly better grasping ability (Mann-Whitney U-test, $p < 0.001$), (b) significantly higher modularity (Mann-Whitney U-test, $p < 0.001$), (c) significantly more behavioral conservatism (Mann-Whitney U-test, $p < 0.001$), but (d) no significant difference in the amount of pose differences.

mean grasping ability, controller modularity, behavioral conservatism and pose differences for the 100 best robots from these two experiments are reported in Fig. 6.

5. DISCUSSION AND CONCLUSIONS

Fig. 4a indicates that evolving robot morphologies along with RBN controllers produces robots significantly more capable of grasping than robots with fixed morphologies, even though these latter robots are known to be capable of solving the task. Fig. 4b suggests why: evolution is able to find robot morphologies capable of grasping with more modular controllers than the controllers found for the fixed-morphology robots. However, the difference in neural modularity between the fixed- and evolved-morphology robots is not statistically significant.

This suggests that directly selecting for modular controllers may increase the evolvability of the evolving robots, enabling them to react appropriately to new environments which are novel combinations of familiar percepts. However, when both grasping and neural modularity are selected for performance decreased: exceedingly modular controllers are discovered (Fig. 5b) but at the expense of grasping ability: regardless of whether robots have fixed or evolved morphologies, few reach even the third or fourth environment (Fig. 5a).

We hypothesize that this is because evolution tends to maximize the interconnectivity among the hand neurons, and among the arm neurons. Such dense connectivity within neural modules likely makes the addition of new attractors corresponding to new useful hand (or arm) poses difficult. Indeed, the inability of dense networks to add attractors while preserving existing ones has been reported in [18]. An alternative explanation is that the multiobjective method employed here may have impaired this set of runs. In future work we will compare these results against those produced by a single-objective optimization method in which graping ability and neural modularity are multiplied together.

Returning to robots evolved only to grasp, there is another feature of the successful robots that was signficantly higher among the evolved-morphology robots than among those with fixed modularity, even though it was not selected for: the conservatism of their behavior (Fig. 4c). When conservatism is directly selected for along with grasping (Fig. 6), significantly better grasping is achieved (Fig. 6c), even though, due to the additional fitness objective of conservatism, selection pressure acting directly on grasping is weaker. Indeed, robots with evolved morphologies and controllers that were selected for grasping and conservatism yielded significantly better grasping that the robots produced by experiment sets two through six (Fig. 7).

The robots produced by selecting for grasping and conservatism also achieved significantly higher neural modularity (Fig. 6b), even though this trait was not directly selected for. These results, taken together, support the hypothesis outlined above: Under these particular pressures, and given the ability to mould the body of robots, evolution seems better able to find morphology and controller pairs that enable successful grasping such that the arm only attends to changes in object position and the hand only attends to changes in object size. Evolution accomplishes this by increasing neural modularity in such a way that arm and hand motor neurons become more independent (Fig. 6b).

This modularity then presumably enables robots to rapidly evolve in new environments, because the evolving arm modules and hand modules both react appropriately to a familiar object position and object size, respectively, even if those percepts appear in a novel combination. This can be seen

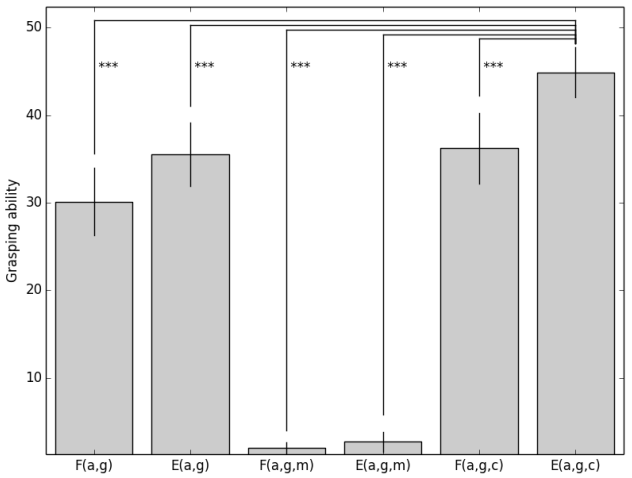

Figure 7: Comparative grasping abilities of evolved robots. $F(x)$ and $E(x)$ denote robots with fixed and evolved morphologies, respectively. (a, g) denote genomes evolved for age and grasping ability; (a, g, m) denote genomes evolved for age, grasping ability and neural modularity; (a, g, c) denote genomes evolved for age, grasping ability and behavioral conservatism.

in the evolved robot shown in Fig. 3: this robot employs the same hand pose in environment 4 as the one it evolved to succeed in environment 3, and it also employs the same arm pose in environment 4 as the one it evolved to succeed in environment 2.

This work thus suggests a new reason for why roboticists should pay careful attention to the design of a robot's mechanical structure. Even if a robot is designed in such a way that it is known to be capable of solving specific instances of a task, it may not be evolvable: poor design choices may inadvertently require additional learning or the evolution of new behaviors when presented with any new instance of the task. This work may also have implications for neuroscience. There is evidence that distinct regions of the brain come to reflect discrete structure out in the world [15]. However, brains also strike a balance between localized and distributed representations [11], which we also see in the modular but not extremely modular nature of our robot's evolved controllers. Finally, the demonstrated ability of the robots here to recognize novel combinations of familiar percepts could pave the way to robots capable of more flexible cognition through the establishment of compositional symbol grounding from motor patterns [7].

In future work we plan to scale up the complexity and physical realism of the robot and its task environment. Specifically, we wish to investigate whether in more challenging domains behavioral conservatism will be of increasingly utility such that it does not have to be directly selected for. Also, we plan to study robots which must act on their environment to draw out the invariant features that modules should evolve to deal with independently.

6. ACKNOWLEDGEMENTS

We thank Jodi Schwarz for many thoughtful discussions that contributed to the fruition of this project. This work was supported by NSF Awards PECASE-0953837 and INSPIRE-1344227 and DARPA award MSEE-FA8650-11-1-7155.

7. REFERENCES

[1] J. C. Bongard. Spontaneous evolution of structural modularity in robot neural network controllers. In *Procs of GECCO*, pages 251–258. ACM, 2011.

[2] J. C. Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–83, 2013.

[3] R. Calabretta, S. Nolfi, D. Parisi, and G. P. Wagner. Duplication of modules facilitates the evolution of functional specialization. *Artificial Life*, 6(1):69–84, 2000.

[4] J. Clune, B. Beckmann, P. McKinley, and C. Ofria. Investigating whether HyperNEAT produces modular neural networks. In *Procs of GECCO*, pages 635–642. ACM, 2010.

[5] J. Clune, J.-B. Mouret, and H. Lipson. The evolutionary origins of modularity. *Procs of the Royal Society B: Biological sciences*, 280(1755):20122863, 2013.

[6] C. Espinosa-Soto and A. Wagner. Specialization can drive the evolution of modularity. *PLoS Comp Biol*, 6(3):e1000719, 2010.

[7] A. Greco and C. Caneva. Compositional symbol grounding for motor patterns. *Frontiers in Neurorobotics*, 4, 2010.

[8] F. Gruau. Automatic definition of modular neural networks. *Adaptive Behaviour*, 3:151–183, 1994.

[9] N. Kashtan and U. Alon. Spontaneous evolution of modularity and network motifs. *PNAS*, 102(39):13773, 2005.

[10] J. Long. *Darwin's Devices*. Basic Books, 2012.

[11] B. Z. Mahon and A. Caramazza. What drives the organization of object knowledge in the brain? *Trends in Cognitive Sciences*, 15(3):97–103, 2011.

[12] M. E. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[13] S. Nolfi. Using emergent modularity to develop control systems for mobile robots. *Adaptive Behavior*, 3–4:343–364, 1997.

[14] S. Nolfi and D. Floreano. *Evolutionary Robotics*. MIT Press, Boston, MA, 2000.

[15] T. Orlov, T. R. Makin, and E. Zohary. Topographic representation of the human body in the occipitotemporal cortex. *Neuron*, 68(3):586–600, 2010.

[16] M. Schmidt and H. Lipson. Age-Fitness Pareto Optimization. *Genetic Programming Theory and Practice VIII*, pages 129–146, 2011.

[17] K. Stanley, D. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[18] C. Torres-Sosa, S. Huang, and M. Aldana. Criticality is an emergent property of genetic networks that exhibit evolvability. *PLoS Comp Bio*, 8(9):e1002669, 2012.

[19] G. Wagner, M. Pavlicev, and J. Cheverud. The road to modularity. *Nature Reviews Genetics*, 8(12):921–931, 2007.

[20] Y. Yamashita and J. Tani. Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Comp Bio*, 4(11):e1000220, 2008.